IMISketch: an interactive method for sketch recognition

Achraf Ghorbel, Eric Anquetil, Jean Camillerapp, Aurélie Lemaitre

IMISketch: an interactive method for sketch recognition

Achraf Ghorbel[a], Eric Anquetil[a], Jean Camillerapp[a], Aurélie Lemaitre[b]

[a] *INSA de Rennes, UMR IRISA, Campus de Beaulieu, F-35042 Rennes*
Université Européenne de Bretagne, France
[b] *UMR IRISA, Campus de Beaulieu, F-35042 Rennes*
Université de Rennes 2, France

Abstract

In this paper, we present a new generic method for an interactive interpretation of sketches to avoid a fastidious verification phase. After a preprocessing phase in which we extract a set of primitives, the interpretation process consists of an interactive analysis. The analyzer is based on a competitive breadth-first exploration of the analysis tree. As opposed to well known structural approaches, this method allows to evaluate simultaneously several possible hypotheses of recognition in a dynamic local context of document. The decision process is able to solicit the user in the case of strong ambiguity: when it is not sure to make the right decision. The user explicitly validates the right decision. While, in practice these approaches often induce a large combinatory, this paper presents optimization strategies to reduce the combinatory. The goal of these optimizations is to have time analysis compatible with user expectations. These strategies have been integrated into both preprocessing and analysis phases. To validate this interactive analysis method, several experiments are reported in this paper on off-line handwritten 2D architectural floor plans.

Email addresses: achraf.ghorbel@irisa.fr (Achraf Ghorbel), eric.anquetil@irisa.fr (Eric Anquetil), jean.camillerapp@irisa.fr (Jean Camillerapp), aurelie.lemaitre@irisa.fr (Aurélie Lemaitre)

Keywords: sketch recognition, breadth-first exploration, interactive recognition, 2D architectural floor plans, Kalman filter

1. Introduction

Nowadays, digital documents are becoming more and more omnipresent in our life. Many reasons, such as the flexibility provided by digital processing, have led to transform handwritten documents to digital ones. In this context, people are working on mapping technical paper documents, like architectural floor plans, to digital ones. We aim at offering a complete and homogeneous solution to unify paper document recognition and pen-based composition (for instance: with Tablet PC). We present the IMISktech system : Interactive Method for Interpretation of Sketches. The input of this system is a scanned image of handwritten architectural plan and after interpretation the output is its digital version. This method is the result of four years of research that leads to several scientific publication [14, 16, 17, 15]. This paper is a synthesis of this method. We focus on the extraction of primitives that feed the analyzer to optimize the management of combinatory.

We have identified two major approaches for document analysis: syntactic and statistical approaches. Choosing one of these two approaches depends on the document type.

The syntactic approaches [6, 8, 29, 12, 21, 22] lean on prior knowledge of the document structure to drive the analysis. They are often based on visual languages for describing this knowledge and generating the analyzer. However, syntactic methods have difficulties to incorporate the uncertainty.

The statistical approaches [25, 30, 32] provide a better ability to incorporate uncertainty and usually lack the ability to convey the hierarchical structure of the document. The use of statistical approaches needs a wide learning on a homogeneous and labeled base. Each type of approach has advantages and drawbacks. The interpretation of handwritten structured documents needs on the one hand an approach that retains its structure, ie a syntactic approach, and on the other hand an approach that provides a better ability to incorporate uncertainty, ie a statistical approach.

In this work, we design a complete system for the sketch interpretation: IMISketch[1]. One of the main originalities of IMISketch is to avoid an *a posteriori* verification phase by soliciting the user. After a preprocessing phase in which primitives of the structured document are extracted, the system is characterized by an interactive analysis phase. The analyzer (referred as IMISketch) uses a new syntactic approach based on an interactive and lazy interpretation of the document. Unlike the classical syntactic approaches, IMISketch does not always select the first or the best found hypothesis. The associated analysis process is able to take into account the uncertainty.

Thanks to the interactivity, the user can be solicited, if needed, by the analyzer to raise ambiguities of recognition [17] i.e. to choose between two or more possible hypotheses or to enrich the *a priori* knowledge of the system [14]. In fact, the user participation has a great impact to avoid error accumulation during the analysis step. To detect ambiguities, we should adopt a method based on breadth-first exploration. Like all the analysis methods based in breadth-first exploration, this approach can induce a large

[1]Interactive Method for Interpretation of Sketch

combinatorics. This combinatorics mainly depends on the quality of primitives extracted from the image and the manner to analyze them. In this paper we propose some optimizations to reduce it, by addressing these two points. These optimizations are introduced from the phase of segmentation to the analysis. They will lead to the new system IMISketch+.

The complete system can be applied to off-line documents (image), as illustrated in this paper (Figure 9(a)), as well as on-line or vectored documents.

In the state of the art, one interesting generic approach is the LADDER [19] [18] system which has been proposed by Hammond and Davis for interpreting *a posteriori* or on the fly on-line handwritten documents. LADDER language has been exploited for the design of various systems of interpretation of structured documents, such as UML [18], electrical diagrams [2] or complex graphs [20]. Also Plimmer proposed InkKit that is a framework and a toolkit to recognize complex components [31] [13]. In addition VR Sketchpad [27] is a pen-based computing environment for inputting and locating 3D objects in a virtual world.

Unlike these methods, our method interprets off-line handwritten structured documents. It has been tested on 2D architectural floor plans. The specific task of floor plan analysis has been addressed for more than twenty years. Lladós [26] proposed a method for understanding hand drawn floor plans using subgraph isomorphism and Hough transform. Aoki [3] proposed also a method for interpreting a hand-sketched floor plan. This method focuses on understanding the hand sketched floor plan and converting it into a CAD representation. Also, Ahmed [33] proposed an analysis method specified in printed architectural floor plans.

Contrary to these methods that can require a fastidious *a posteriori* verification phase, IMISketch system attempts to avoid this phase by integrating the user during the analysis process.

The recognition of a structured document using a structural approach needs an a priori description. The modeling of structured documents differs from one type to another. Several techniques allow the document description. Yamamoto [35] and Bunke [5] use classical one-dimensional grammars. Other techniques are used to model two-dimensional documents. Fahmy [11] and Bunke [4] offer grammar graphs. These grammars have been widely used in the various communities for interpreting off-line documents such as mathematical formulas. Despite graph grammars offer a very expressive mechanism for pattern recognition, these grammars have their limitations. They are expensive to implement and difficult to handle by the developer, especially when the productions become numerous. These graphs are also poorly adapted to deal with uncertainty.

Our goal is to analyze documents of different kinds such as handwritten documents. To overcome this problem, we adopt context-driven constraint multi-set grammars (CD-CMG), designed for *on-line* recognition [28] associated with a scoring approach based on the fuzzy logic theory. The main contribution of our work is to modify and to extend this formalism to design an interactive analyser for *off-line* recognition. This strategy allows to solicit the user in the case of strong ambiguity and avoids the fastidious a posteriori verification task to find and correct the remaining interpretation errors.

The remaining of the paper is organized as follows. In the section 3(a), we introduce the architecture and the basic principles of IMISketch method. The phase of primitive extraction is described in section 3. Section 4 presents the concepts that are linked with an interactive breadth-first analysis. In section 5 implementation and optimization of IMISketch analyzer are presented. Experimental results on interpretation of images of 2D handwritten architectural floor plans are reported in section 6 and finally, section 7 concludes the paper.

2. Interactive analysis stages

In this section, we summarize the different steps of treatment to ensure the recognition of a handwritten structured document (cf. Figure 1). The first step is the segmentation process. This step is purely off-line (i.e. without user interaction). The aim of this phase is to extract all the basic primitives that will be used to analyze the document. In the context of sketch recognition, the segmentation process consists in extracting handwritten strokes as a set of segments. This part is detailed in section 3. The second main step is to analyze these primitives and to compose them according to their structural arrangement in the document to identify the symbols. Our analyzer is made of two associated key processes: the recognition of the document structure is managed by **the grammatical analyzer** that drives the calls of **symbol classifiers** to evaluate a fuzzy scoring for each hypotheses. For instance, in architectural plans, structural recognition detects walls, opening, etc., and the fuzzy classifiers identify the opening to window, door, etc.

The user can be solicited during the analysis process in case of ambiguity detection. This approach is generic and needs some *a priori* knowledge at the structural level by defining a grammar that will describe the document structure and at the classifier level to identify the symbols. This knowledge corresponds to the specific part of the analysis system.

We detail in the following sections the preprocessing (section 3) and the analysis phases (section 4).

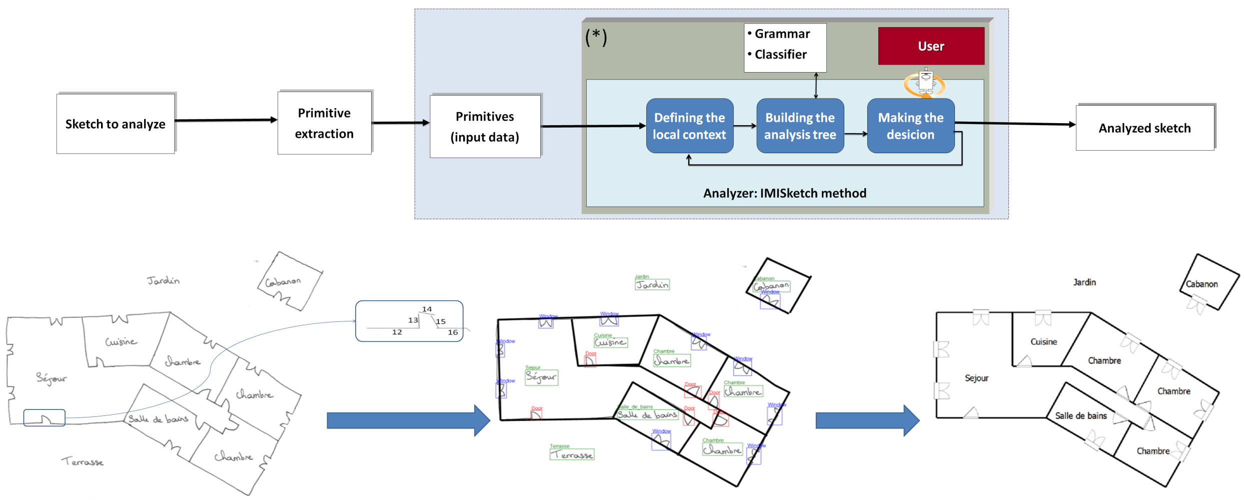


Figure 1: Synthetic scheme for structured documents interpretation

3. Preprocessing phase: primitive extraction

3.1. Related works

The recognition of architectural plans has already been studied in particular by Dosch et al [10]. In this kind of plans, segments, representing the walls, are primitives widely used. In these works, most of the analyzed plans have been drawn with a ruler (or CAD software), consequently, segments are really straight. In [23], Hilaire proposed a method that improves the method of Dosch. The originality of his work comes from the segmentation process of the skeleton. This aspect is very important in the context in which drawings are crossing. His paper contains an important bibliography. Hilaire, as Dosch, works on printed documents, but our aim is to process handwritten documents. In such documents, walls are not straight lines and the representation of doors and windows use arcs of a circle which are sometimes very roughly drawn.

In [7], Chang proposes to carry out the vectorization of handwritten documents by Bezier curves. This approach has been designed for the vectorization of cartoons, but it does not lead to an easy representation to be used for a later interpretation of symbols like pieces, doors, windows, etc.

In [9] de Brucq uses a Kalman filter to decompose handwritten strokes into straight lines and arcs of circle. These primitives are well adapted for architectural plan interpretations, but the system works with an on-line representation of the drawing.

3.2. Our method

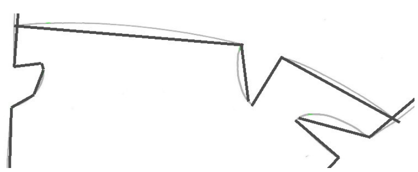
To extract lines in off-line sketch documents, we propose to adapt the method we presented in [24]. It uses a Kalman filter.

A Kalman filter is a prediction verification process that provides an estimate of a model from observations. In the case of the drawing extraction, our model is based on three values: the thickness, the position and the local slope of the line. When the drawing is relatively regular, the Kalman filter can use these three values to predict the next position and update the model from the pixels of the image. In areas that include intersections, the observed thickness is not consistent with the predicted thickness, but it is possible to use the prediction ability of the filter to pass through these regions. In a region where the direction of the line changes rapidly, it is not possible to find black pixels in the predicted position and the follow-up of the drawing stops. Equations and more details can be found in [24].

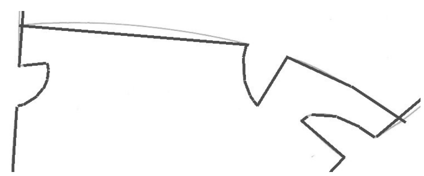
For sketch documents the most important parameter for the Kalman filter is the one that defines the variability of the slope. With a large value (0.1, i.e. a 10% variation of the slope) a curved section of drawing is detected as a single segment joining its two ends (Figure 2(a)). With a very low value (0.005) the model allows a very little variation of the slope, and thus many small segments are detected in the same curved section (Figure 2(b)). We note, in this example, that the symbols representing opening (door or window) are more precisely represented in Figure 2(b) (precise decomposition) than in Figure 2(a) (rough decomposition). A precise decomposition increases the number of segments extracted from the image, however this decomposition can be very useful when using a classifier to recognize symbols.

To fix ideas, the full plan from which is extracted the Figure 2 is represented by 144 segments for the rough decomposition and 177 segments for the precise one. This increase of the number of segments is mostly due to small segments in the opening regions. We must notice that a little region with many small segments will produce a combinatorics explosion of the analyze time. This effect is pointed out later in the section 6 and Figure 9(c).

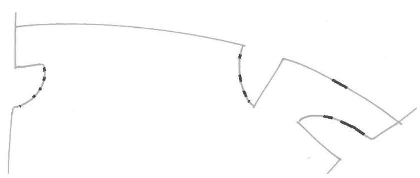
To overcome this dilemma, i.e. a precise representation but a low combinatorics, we propose to extract only a precise decomposition and to build spatial relationships between the segments. Thus, it is important to remember if a segment is a curvilinear extension of another one. During the



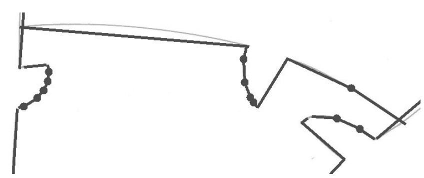
(a) Rough decomposition of sketch



(b) Fine decomposition of sketch



(c) Common part of two segment is shown dark



(d) Representation with segments and lines (circle denotes connection between consecutive segments)

Figure 2: Extraction of primitives: Original drawing in light gray, primitives shown in dark black

detection of segments by the Kalman filter, it is possible to detect that a part of the drawing belongs to two different segments (Figure 2(c)). In this case, we remove this common part from one of the two segments and memorize that their is a curvilinear link between the extremities of these two segments. The small circles in Figure 2(d) represent a connection between two segments. Each set of segments attached to one another corresponds to a polygonal approximation of the curved parts of the drawing (for instance an opening).

Thus, there are two types of primitives extracted from an image: simple segments and polygons. We transmit the connection information so that the interpretation process can directly use simple segments and polygons as primitive, or can decide to split a polygon into segments according to some structural information like the size of the polygon or criterion of collinearity. The use of this dual representation on one hand allows a more precise characterization of curve sections and on the other hand limits the combinatorics during the interpretation process.

4. Interactive breadth-first exploration

In this section, we present the analyzer by first describing its main characteristics. Then, we detail the different steps of the internal analysis. The description of this method is followed by a concrete example in the next section (section 5).

4.1. Analyzer characteristics

The interpretation strategy of structured documents is driven by the *a priori* knowledge we have on the structural rules of the application domain.

These structural information are formalised by a visual language from which we can automatically develop a breadth-first interactive analyser. For handwritten architectural plans, the interpretation of a primitive takes into account the neighbouring objects. Moreover, as we work in a two-dimensional context (images), it can induce a large combinatorics. To overcome this complexity problem, we propose to guide our breadth-first exploration using a spatial contextual focus.

This idea is close to the well known LL(k) analysis, where reading the next k tokens enables to choose without ambiguities which rule must be applied. In the same manner, in our two-dimensional analysis, we have to limit the number of token k to explore, i.e. the depth of the analysis. Consequently, contrary to the LL(k) analysis, the exploration of the following tokens does not allow to take a unique decision on the rule to apply, because we voluntarily limit the value of k. Moreover, sometimes, the grammar is not LL(k) for any k and the analyser meets ambiguities. In those two cases, the process can not be sure to take itself the decision, and may hesitate between several hypotheses. In order to validate the right decision, we propose an analysis process that can, through its decision process, take the right decision or solicit the user in case of ambiguities. This analyzer is based on the following characteristics:

- the expression of a priori structural knowledge of the document through a visual language based on production rules;
- a two-dimensional descending breadth first analysis;
- a spatial contextual focus of the exploration to limit the combinatorics;

- the formalization of uncertainty by the attribution of scores to each hypothesis, represented by a tree analysis branch;
- some user solicitaions: if the ambiguities can not be resolved in the local context in an automatic manner, the user will be solicited by the analyzer to resolve the ambiguity.

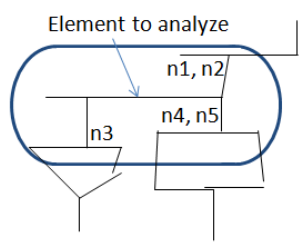
These characteristics have been defined to ensure a good interaction between the process of analysis and the user. This interactivity allows in particular to avoid an *a posteriori* verification phase, which can become fastidious on complex documents. Indeed, the user participation, on the critical phases of the analysis of the document, has a great impact to avoid error accumulation during the analysis step and overcomes the combinatorics due to the sketch complexity.

4.2. Steps of the analysis

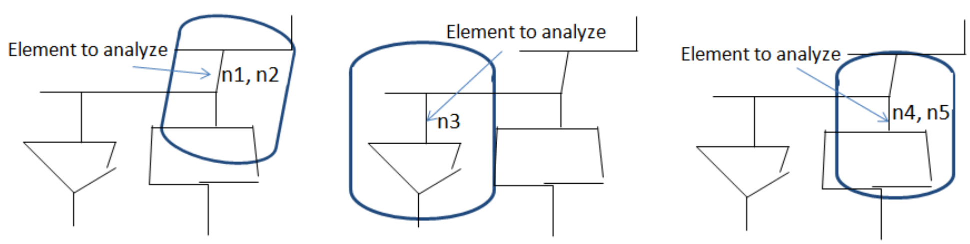
The analyzer tries to match the set of primitives contained in the document with the structure model defined by the two-dimensional production rules. The interactive breadth-first analysis process consists of three stages: 1) defining the local context, 2) building the analysis tree and 3) making the decision. The inner part(*) of Figure 1 illustrates these three phases and the relationships between them.

4.2.1. Defining the local context

The recognition of a given primitive depends on its neighbourhood in structured documents. The analyzer begins by defining a spatial contextual focus that aims to limit the combinatorial exploration due to the breadth-first exploration of tree analysis. The structured document requires a two-dimensional context. This two-dimensional local context is defined for an analysis tree as the maximum distance between the elements of the root and the elements of any leaves. The choice of the size of the local context depends on the application domain. For example, to interpret an architectural plan, we suggest a local context with a size corresponding to the maximum size of an opening (door, window, etc.). Figures 3(a) and 3(b) show the shifting of the local context in two consecutive steps. Once the local context is set, we go to the building of the analysis tree stage.



(a) Local context at step s



(b) Local context at step s+1

Figure 3: Dynamic adaptation of local context around the element to analyse

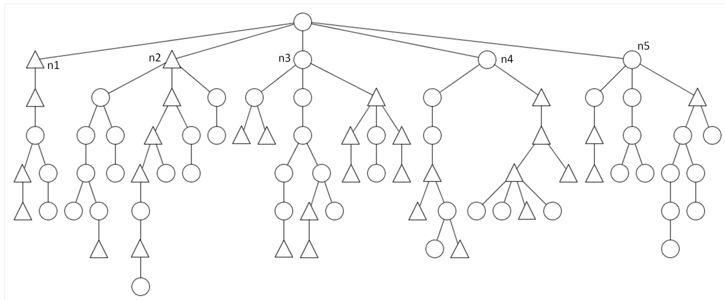
4.2.2. Building the analysis trees

In this stage, the analyzer explores all the possible hypotheses of interpretation in the spatial context using a set of two-dimensional rules that describe the structure of the document. Each primitive can be interpreted in several ways. Each node or leaf is the application of a production rule deduced from the previous node. Every leaf or node of the tree has a score calculated from both its local score and the score obtained from the preceding nodes. Every score determines the adequacy degree to validate a production. It is calculated from each rule. The production score can also be deduced from a classifier. Each branch (hypothesis) is characterized by a score.

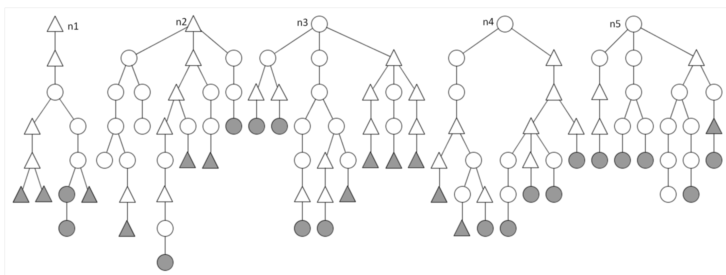
The breadth-first exploration using a local context of the IMISketch method is illustrated in Figures 4(a) and 4(b). This exploration generates some combinatorics. Consequently, we describe in section 4.2.2.a a new algorithm for constructing analysis trees for reducing the combinatorics.

4.2.2.a. Our new IMISketch+ method of tree construction

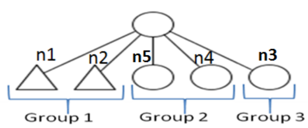
Each analysis tree characterizes the elements to interpret in the defined local context. Each root is the production rule that would consume this primitive. The number of analysis trees corresponds to the number of possible interpretations for the current primitive. The analysis tree then contains a set of complete or incomplete objects. An object is called complete if and only if this object can be found in the final result of the document interpretation. For example, in the case of the architectural plans, the complete objects can be walls, doors, windows... An incomplete object is an object that is not complete, but rather a part of complete object, i.e, we do not find it in the



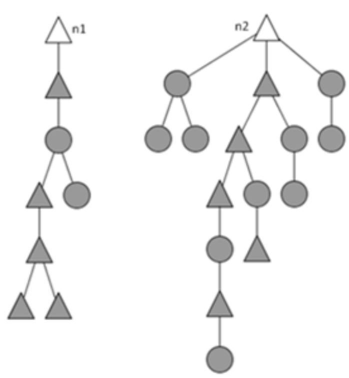
(a) IMISketch method at step s



(b) IMISketch method at step s+1



(c) IMISketch+ at step s. The direct son of the root are grouped by consumed elements



(d) IMISketch+ at step s+1. Only the roots (n1, n2), belonging to the same group (Group 1) are developed

Figure 4: IMISketch Vs IMISketch+: transition between two successive analysis trees. Nodes modeled by circles refer to complete object. Nodes modeled by triangles refer to incomplete objects. The grayed leaves indicate the new applied production rules after movement of the local context.

final result of the interpretation. These objects can lead to complete objects (furniture, openings).

Despite the use of a local context, the problem of combinatory and therefore computing time, is an important problem that must be addressed to meet the criteria of acceptability and usability of this type of interactive analysis with a user. The combinatory is mainly due to the number of applied production rules in the local context.

In fact, the development of certain nodes in tree analysis is useless. That is why, we present a new approach to build the tree analysis in order to only develop the useful nodes for making the right decision [16]. This algorithm is described below:

- If the number of analysis tree roots is equal to 1: (Figure 4(c))

    1. limit the development of direct sons of the root. If the root to develop is unique, we can say that only one interpretation is possible for the consumed element. The decision process is then sure to validate the right root. In this case, since we know the root to be validated, we consider unnecessary to build all the tree analysis, and we can limit the tree development to the direct sons. After validation of this root, these direct sons will be the new roots to build.

- If the number of analysis tree roots is greater than 1: (Figure 4(d))

    1. regroup the roots by consumed elements. We will not build all roots but only those that share the same elements. We want

to find the right interpretation in order to consume the element within the root.

2. order these groups by their scores. Each group has a score derived from the roots within it. This score is the score of the best hypothesis (branch) located in the group.
3. develop only roots belonging to the group having the best score.
4. build analysis tree as long as the following conditions are verified:
   - the newly consumed element is in the local context of research;
   - the number of consumed elements in each hypothesis (branch) is below a threshold. In the architectural plans, we note that the number of primitive of an opening can not exceed 10 primitives. For this we fix this threshold to 10.
   - the number of complete elements belonging to a branch is less than a second threshold. We can fix a threshold to 3 for the architectural plans.

Figures 4(d) and 4(c) illustrate an example of building the analysis trees with our new method (IMISketch+). The local context is not only limited to the distance between primitives but also to the number of complete elements in hypothesis. This optimization can generate a lack of information on hypotheses and therefore a possible ambiguity. But thanks to the interactivity, this insufficiency does not influence on the final result as far as the user may be solicited to validate the right hypothesis.

4.2.2.b. IMISketch+ Vs IMISketch

The aim of this section is to show the improvements deduced from the new algorithm (IMISketch+) for building analysis trees compared to the classic exploration method IMISketch (in which all possible branches are explored). To facilitate this comparison, we present an example of an artificial tree analysis. The goal is to compare, for each step, the number of developed nodes according to the placement of the local context depending on the elements to be analyzed. The transition of an analysis tree to the next analysis tree involves a shift of the local context. This movement allows applying the other productions. In fact, the building of the new analysis trees is not the rebuilding of the whole branches but only the new found productions by shifting the context.

Figure 4(a) shows the result of the exploration according to the IMISketch method. The number of interpretation (nodes) is equal to 80 interpretations. The tree construction based on the IMISketch+ algorithm generates only 6 interpretations (Figure 4(c)). Figures 4(b) et 4(d) illustrate the new analysis trees by moving the local context. After two successive construction steps of analysis trees, we went from 111 interpretations (IMISketch method) to 28 interpretations (IMISketch+), this corresponds to a gain of about 75% of computation time. These optimizations are generic and do not depend on the category of structured document to analyze.

4.2.3. Making the decision

Once the tree is well constructed, we start the decision phase. The role of the decision process is to validate the right hypothesis among a set of competing hypotheses generated with a descending breadth first analysis. Two cases are presented (section 4.2.2): case of a single root and case of several roots.

- Case of a single root: this root is validated.
- Case of several roots: the hypothesis (branch) having the highest leaf score in each tree is considered. These branches are sorted by the score of each leaf. Then we compare the obtained branches. Two cases may appear:
  - Implicit validation: when the analyzer is confident enough to choose the correct root without requesting the user. The analyzer implicitly validates the root which has the branch having the highest score.
  - Explicit validation: when the decision process requires the user to validate the right decision. In practice, if the difference between the branch with the highest score and another branch is below a threshold, called *threshold of confidence* and these two branches are contradictory (at least one joint primitive is not consumed by the same rule production), the user intervention is required.

The decision is not limited only to validate the correct root but can also validate directly a part of the branch (hypothesis), for accelerating the analysis. In general, if the direct son of a node is unique, the validation of this node generates automatically the validation of its direct son. The decision making is summarize in Algorithm 1.

Function Making the decision(right hypothesis : **list of nodes**) : **list of nodes**

    validated-nodes : **list of nodes**;
    validated-nodes.add(root of the right hypothesis) ;
    successor ← validated-nodes.lastElement.successor;
    While (Number of successor == 1) **do**
        validated-nodes.add(validated-nodes.lastElement);
        successor ← validated-nodes.lastElement.successor
    done
    return validated-nodes;

End

Algorithm 1: Decision algorithm

When the correct root is validated, other roots are put on hold and the new roots are either the sons of this root if exists, or the waiting roots otherwise and the analyzer goes back to the first step (defining the local context step). The analysis is complete when no more production rule is applicable.

5. Implementation of IMISketch

In this section, we describe the implementation of our interactive analysis method (IMISketch) and illustrate it on 2D handwritten architectural plans.

5.1. Use of CD-CMG

Given that architectural plans are two-dimensional structured documents, for modeling our interactive analyzer, we adopt the two-dimensional grammars. We use the context-driven constraint multiset grammars (CD-CMG) (detailed in [28]). The analysis process is driven by the context that involves a significant decrease in the complexity of the analysis process. The score calculated by each production (node) is due to preconditions and constraints of the rule production [28]. Equation 1 defines the manner that we calculate the score for each production. The use of the square root is a normalization using a geometric average.

$$\rho_P = \sqrt{\mu_{preconditions} \cdot \mu_{constraints}} \tag{1}$$

Deducing $\mu_{preconditions}$: the computing of $\mu_{preconditions}$ is simply the fuzzy application of the precondition block. Each DSC [2] is evaluated and the corresponding membership degrees are merged: a fuzzy conjunction (t-norm) is used for an '&', whereas a fuzzy disjunction (t-conorm) is used for an '|'. Once again, the resulting degree is normalized to avoid giving an advantage to productions with less DSC.

Deducing $\mu_{constraints}$: the computing of $\mu_{constraints}$ is based on the same principle than the computing of $\mu_{preconditions}$. It means that each constraint (structural or statistical) must return a membership degree. In practice, defining fuzzy structural constraints is often straightforward and recognition systems we use are based on fuzzy inference systems from which we obtain

[2] DSC is a specific constraint modeling both a location in the document and elements that are awaited in it.

such degrees [34].

Equation 2 determines the degree of adequacy (score) of a hypothesis. |PS| is the number of production in the considered branch (referred PS). ρ_{P_i} is calculated by Equation 1.

$$\rho_{PS} = (\prod_{P_i \in PS} \rho_{P_i})^{\frac{1}{|PS|}} \tag{2}$$

A production rule can call an external classifier to recognize the symbols. In our application context, the classifier is used for the recognition of the types of opening (eg door, window, etc.). Each recognition is associated to a score. This classification system is based on first-order Takagi-Sugeno (TS) fuzzy inference system [1].

5.2. Impact of the primitives

To demonstrate the interest of using the polygons (cf. section 3) to reduce the combinatorics, we compare a first set of production rules for interpreting primitives composed only with segments (Table 1(Ps-P5)) and a second set of rules dealing with two kinds of primitives : the segments and polygons. This second set consists of the same rules enriched by three new rules that allow to analyze polygons (Table 1(P6-P8)).

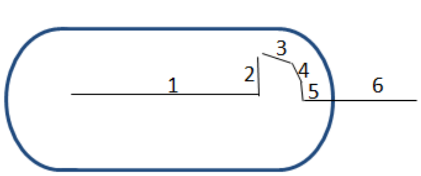
By applying the production rules described in Table 1 to a set of segments extracted from an architectural plan (Figure 5(a)), we obtain the analysis tree illustrated in Figure 5(c). In the next step, the local context is shifted (Figure 5(b)). We then obtain tree analysis described in Figure 5(d). Despite this example is quite simple, these two analysis trees contain 28 nodes.

Now, we integrate the new polygon primitive. We add production rules illustrated in Table 1(P6-P8). To interpret the segment 1, the analyzer de-

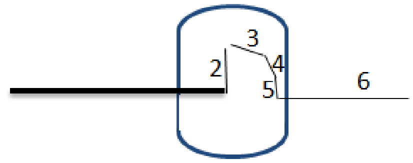
Table 1: Example of production rules used for the architectural plans using **segments** (Ps-P5) and using **polygons**(P6-P8)

Name	**Created elements**	**Consumed elements**
Ps	Starting wall	longest **segment** in the document
P1	Wall	**segment** at the end of an other *wall*
P2	Sequence	**segment** at the end of an interpreted *wall* or *sequence*
P3	Opening and wall	a *sequence* and two collinear *walls* or a *sequence* and *wall* and a **segment** which are collinear
P4	Door	an *opening*
P5	Window	an *opening*

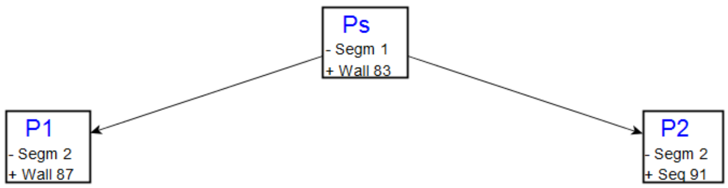
Name	**Created elements**	**Consumed elements**
P6	Wall	**polygon** at the end of an other *wall*
P7	Sequence	**polygon** at the end of an interpreted *wall* or *sequence*
P8	Opening and wall	a *sequence* and *wall* and a **polygon** which are collinear



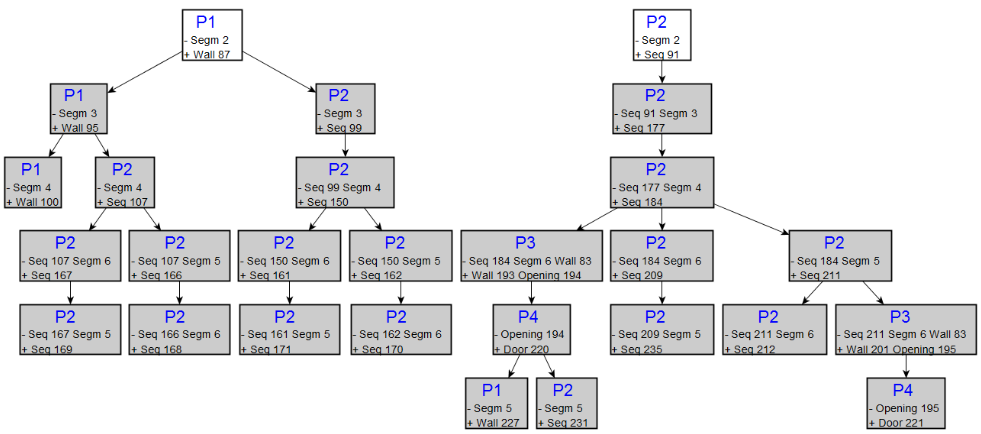
(a) Local context at step n (centered on the segment 1)



(b) Local context at step n+1 (centered on the segment 2)



(c) Analysis tree at step n



(d) Analysis tree at step n+1. The grayed leaves indicate the new applied production rules after shifting the local context.

Figure 5: Position of the local context (box) during the analysis and the associated analysis trees. The primitives only consist of segments.

termines the local context (Figure 6(a)) and builds the tree analysis shown in Figure 6(c). In the next step, the segment 2 will be interpreted, the local context associated with this segment is illustrated in Figure 6(b) and the new tree analysis is described in Figure 6(d). Thanks to the chaining of segments in polygons, we can reduce the combinatorics to 50%. We went from 28 nodes to 14 nodes.

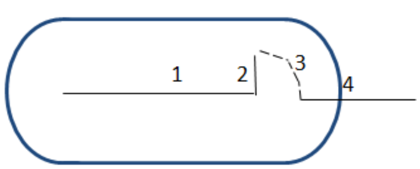
We have shown in this section, the impact of using the chaining of segments on building of the analysis tree. This improvement induces optimizations in the decision process. Moreover, using polygons increases the chances to validate a set of nodes. (cf. section 4.2.3).

The use of polygons as a primitive not only reduces the combinatorics for structural recognition but also guarantees a better shape recognition for obtained symbols through a more accurate representation of its constituents. Indeed, once the symbol is well recognized structurally, all the segments forming the symbol (the segments that are primitive and chained segments forming the polygon) will be forwarded to the classifier to label the symbol. Indeed, once the symbol is well structurally recognized, all the segments of symbol (the segments that are primitive and chained segments forming the polygon) will be transmitted to the classifier to label this symbol.

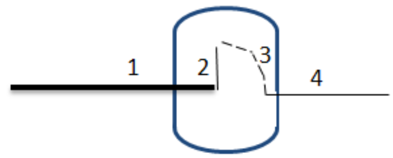
Once the tree analysis is explored, the analyzer starts the final phase: the decision making. In this phase, illustrated in the next section, the user may be solicited for removing ambiguities.

5.3. Ambiguity cases

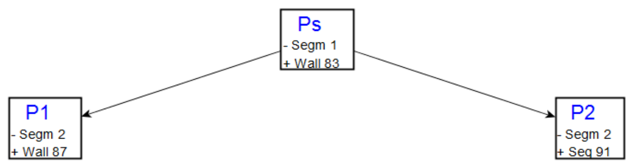
In this section, we show an example of ambiguity which requires a user solicitation. The aim is to analyze the set of primitives extracted from an



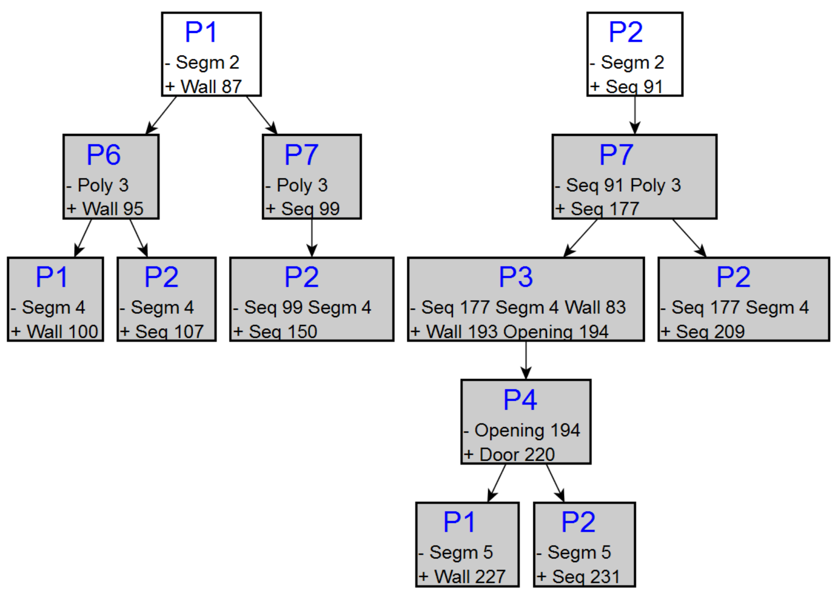
(a) Local context at step n (centered on the segment 1)



(b) Local context at step n+1 (centered on the segment 2)



(c) Analysis tree at step n



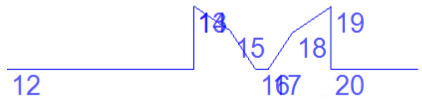
(d) Analysis tree at step n+1. The grayed leaves indicate the new applied production rules after shifting the local context.

Figure 6: Position of the local context (box) during the analysis and the associated analysis trees. The primitives consist of segments and polygons. The dotted line (primitive 3) means the polygon.

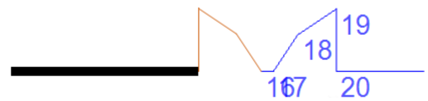
architectural plan (Figure 7(a)). At the step illustrated in Figure 7(b), the decision process decides to report the decision on the user because the two branches in competition are contradictory and the difference between the two scores is lower than the threshold of confidence. The decision process launches an interaction with the user (Figure 8), and proposes to validate the first hypothesis. If the user validates this hypothesis, the document will be recognized as two doors (Figure 7(c)). If the user declines the first hypothesis, the system implicitly validates the second hypothesis (Figure 7(d)) and the interpreted document will be made of one window. We note that the interface is basic for the first version; we have not focused on the HCI design but on the fact to interact with user. We focus on *"when"* the interpretation process solicits the user. *"How"* to interact will be the subject of our future work.

6. Experimental results

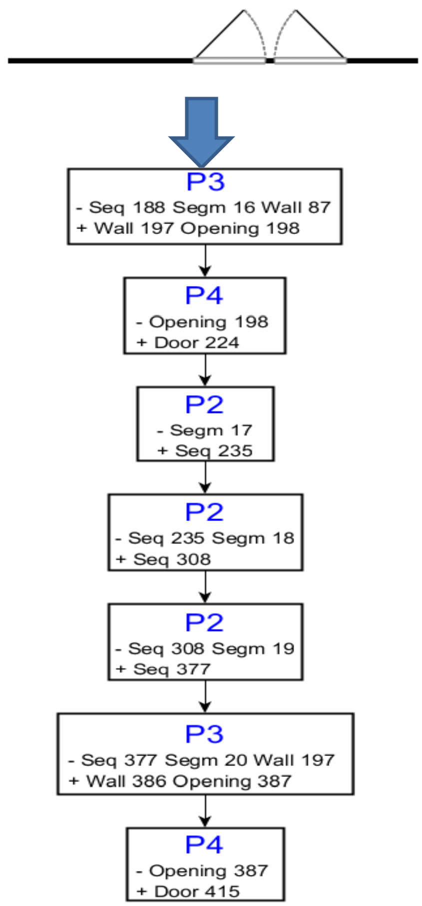
In this section we report different results obtained with the complete interactive recognition system. The results are focused on the impact of the presented optimizations : the use of the polygon primitive and the building strategy of tree analysis in terms of recognition rate and of computing time. For this reason, we propose three versions of IMISketch (Figure 10). The first one (referred as IMISketch) explores all the hypotheses (branches) of the tree analysis in a local context with a set of primitives that contains only segments (without polygons). The second version (referred as IMISketch+(Seg)) takes into account the new algorithm to build the analysis tree keeping the same set of primitives (only segments) as in the first version. The last version (referred as IMISketch+(Seg&poly)) incorporates the both optimizations:



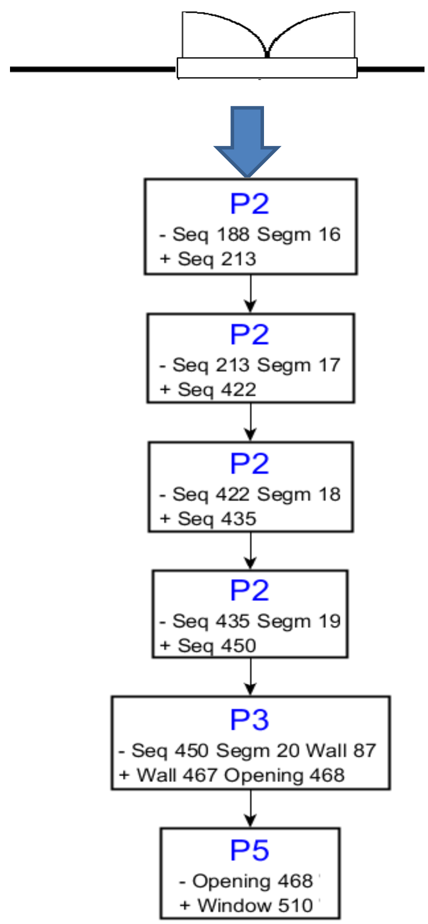
(a) Primitives of the document to analyze



(b) Interface step



(c) The first hypothesis: transforming a set of primitives into a wall and two doors



(d) The second hypothesis: transforming a set of primitives into a wall and a window

Figure 7: Example of ambiguity between two hypotheses

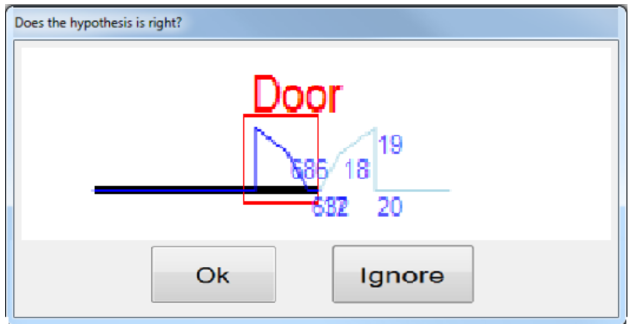


Figure 8: User interface: the analyzer proposes a user to validate the hypothesis described in Figure 7(c). If the user don't validate this hypothesis, the analyzer will validate the hypothesis implicitly described in Figure 7(d)

the use of the polygon primitive and the new algorithm to build the analysis tree (Figure 10).

The experiments were carried on 69 handwritten architectural plans of varying complexity drawn by around ten different people. Each architectural plan is composed of a set of walls, doors, windows and sliding windows. The databases contains 2641 walls, 555 doors, 377 windows and 401 sliding windows. Some examples of architectural plans are illustrated in Figure 9(a). For each architectural plan, we compare, for the three versions (IMISketch, IMISketch+(Seg), IMISketch+(Seg&poly)), the final interpretation result and the obtained computing time (Figure 9).

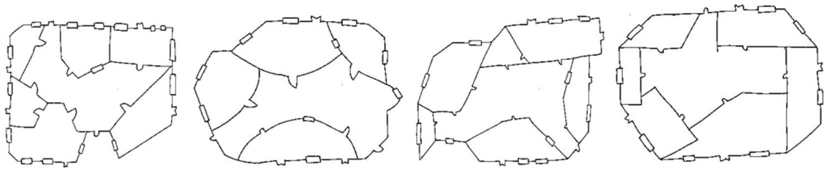
Concerning the recognition rates (cf. Figure 9(b)), we note that the optimizations performed to IMISketch do not affect system performance. Indeed, we observe a very light difference between IMISketch, IMISketch+(Seg) and IMISketch+(Seg&poly) in terms of recognition. With the user solicitation, during the analysis, we obtain 97.96% with IMISketch, 98.13% with

IMISketch+(Seg) and 97.93% with IMISketch+(Seg&poly) (Figure 9(b)).

This can be explained as the hypotheses proposed by the three methods are not the same. IMISketch produces more hypotheses than IMISketch+(Seg) and IMISketch+(Seg) generates more hypotheses than IMISketch+(Seg&poly). The number of competing hypotheses is more important in IMISketch than IMISketch+(Seg&poly). This might suggest that there are more chance to have the right hypothesis with IMISketch, but the generated confusions are also potentially more numerous. In the end, the results in terms of recognition performance are very comparable.

Moreover, the user solicitation decreases from 6 interventions in IMISketch to about 4 interventions in IMISketch+(Seg&poly)(Figure 9(b)). This decrease shows a reduction in the competing hypothesis that can lead to ambiguities. This is due to the polygons which has reduced the number of nodes per branch and therefore the conflicts between hypotheses. The obtained errors (about 2%) are due either to poor calibration of the local context, or the badly drawing of certain symbols.

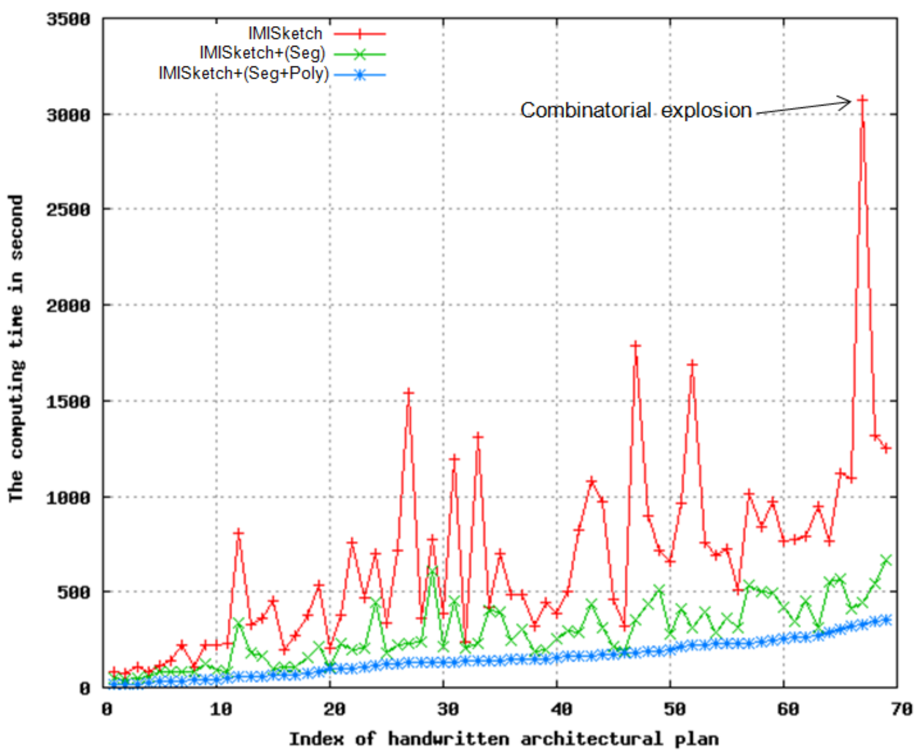
The details of computation time per image between the three versions (IMISketch, IMISketch+(Seg) and IMISketch+(Seg&poly)) are shown in Figure 9(c). The images are ordered by the computing time according to IMISketch+(Seg&poly). We improve the computation time from about 11 minutes of average computing time per image (with classical IMISketch) to about 4 minutes 43 seconds (with IMISketch+(Seg)), i.e. a gain of 57%. IMISketch+(Seg&poly) further reduces computation time to attain about 2 minutes 31 seconds, which presents a total gain of 77%. Depending on the complexity of the plans, the gain of computing time can reach 90% per image



(a) Examples of architectural floor plans

	IMISketch	IMISketch+(Seg)	IMISketch+ (Seg+poly)
Types of primitive	Segments		Segments and polygons
Building the analysis trees	All possible hypotheses in the local context (classical approach)	Using the new approach described in Section 4.2.2	
Number of unrecognized symbols	81	74	82
Recognition rate	97.96%	98.13%	97.93%
Average computation time	11m03	4m43	2m31
Average user solicitation	6.07	5.17	3.59

(b) Characteristics of IMISketch versions and recognition rate on architectural plans per version



(c) Computing time per image showing the disappearance of combinatorial peak using IMISketch+

Figure 9: Experimental results

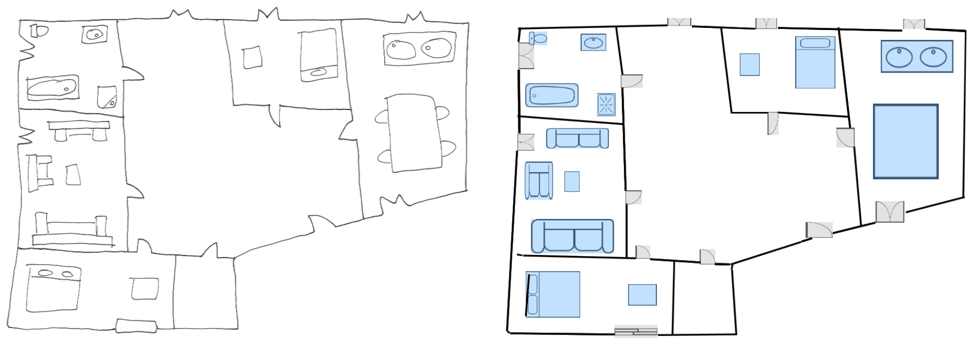
(image with index 67). We note, in Figure 9(c), that the peaks in the analysis time have completely disappeared with the optimized method. Thus, there is no risk for a long wait to interpret a more complex plan. These computation times are obtained in real conditions, i.e. in the presence of a user in the loop of analysis. Today, the average computation time is 2 minutes 31 per image. In a future work, we will try to reduce the computation time to attain around 1 min 30 per image: experimentally, it seems the ideal timing that allows the user to follow in real time the recognition process.

The experimental results are very encouraging. They suggest that it is possible to introduce a descending breadth first analysis by controlling the generated combinatorics. This supports the idea of conceiving an interactive systems for the document recognition. User solicitations, driven by the analyzer, guarantee the obtaining of very high rate of reliability even when considering the treatment of complex documents. the use of the polygon primitive does not have a negative impact on the structural recognition rate, in addition it reduces the number of user intervention during the analysis and also accelerates the computation time.

Current state of IMISketch system

Optimizations of our IMISketch method were carried out to the interpretation of more complex architectural plans containing a dozen furniture types (toilet, table, bed...), 3 types for openings (door, window and sliding window) and walls. Indeed, some experiments on 24 architectural handwritten plans (see Figure 10) show that the structural recognition rate increase from 91% without user solicitations to 96% with user solicitations.

We can notice that the best compromise (recognition rates/user solicita-



(a) Original image (b) Interpreted image

Figure 10: Interpretation of a complex architectural floor plans

tions) is obtained with 12 user solicitations per image: it means that 4% of primitive interpretations require the user solicitation. 49% of user solicitations are useful to take the right decision which is not the best hypothesis proposed by the analyser (more details in [16]).

7. Conclusion

In this paper, we have presented a complete generic method to interpret sketches such as 2D architectural floor plans. This method consists of a preprocessing phase in which we extract useful primitives which constitute the inputs of an interactive analyzer. This analyzer is based on a competitive breadth-first exploration of the analysis tree according to a dynamical local context of the document. The decision process is able to solicit the user in the case of strong ambiguity. Generally, the competitive breadth-first exploration generates combinatorics. Our interactive method integrates an optimization strategy for solving combinatorics. This strategy concerns the preprocessing phase and the analysis phase. The preprocessing phase avoids the aggomeration of primitives in reduced zones of document. At the analysis phase, the strategy is based on a dynamic construction of analysis tree by controlling the depth of each branch following a set of criteria specified in the current local context. It may be noted that this strategy is generic and therefore it could be easily applied to other types of st uctured documents, and other analyzers characterized by a breadth-first exploration. The first tests of this interactive analyzer have been made on 2D handwritten architectural floor plans. Integrating the user in the analysis process is, in our view, a key point to address complex off-line sketch recognition and to avoid an a posteriori verification phase.

Future work will focus on extending the experimental results on large image databases containing more complex architectural plans (integration of furniture, quotes, etc.). We will also validate the criteria of acceptability and usability of the system by doing usage tests that will be conducted in collaboration with experts from the Loustic laboratory (http://www.loustic.net/rennes).

Acknowledgment

The authors would like to thank all the people who took part in the experiments. This work partially benefits from the financial support of the ANR Project Mobisketch.

References

[1] Almaksour, A., Anquetil, E., 2011. Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers. Evolving Systems 2, 25–

33.

[2] Alvarado, C., Davis, R., 2007. Sketchread: a multi-domain sketch recognition engine. In: ACM SIGGRAPH 2007 courses. ACM, pp. 34–es.

[3] Aoki, Y., Shio, A., Arai, H., Odaka, K., 1996. A prototype system for interpreting hand-sketched floor plans. In: Pattern Recognition, 1996., Proceedings of the 13th International Conference on. Vol. 3. IEEE, pp. 747–751.

[4] Bunke, H., nov. 1982. Attributed programmed graph grammars and their application to schematic diagram interpretation. Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-4 (6), 574 –582.

[5] Bunke, H., 1990. String grammars for syntactic pattern recognition. Syntactic and Structural Pattern Recognition. Theory and Applications, 29–54.

[6] Chan, K., Yeung, D., 2000. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. Pattern Recognition 33 (3), 375–384.

[7] Chang, H.-H., Yan, H., 1998. Vectorization of hand-drawn image using piecewise cubic bézier curves fitting. Pattern Recognition 31 (11), 1747 – 1755.

[8] Coüasnon, B., 2006. Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. IJDAR 2006 8 (2), 111–122.

[9] de Brucq, D., Amara, M., Courtellemont, P., Wallon, P., Mesmin, C., 1996. A recursive estimation of parameters of straight lines and circles by an extended Kalman filtering. Application to the modeling of on-line Handwritten drawings. In: International Conference on Signals and Image Processing and Applications (SIPA'96). pp. 213–216.

[10] Dosch, P., Tombre, K., Ah-Soon, C., Masini, G., 2000. A complete system for analysis of architectural drawings. International Journal On Document Analysis and Recognition 3 (2), 102–116.

[11] Fahmy, H., Blostein, D., aug-3 sep 1992. A survey of graph grammars: theory and applications. In: Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on. pp. 294 –298.

[12] Fitzgerald, J., Geiselbrechtinger, F., Kechadi, T., 2007. Mathpad: A fuzzy logic-based recognition system for handwritten mathematics. In: ICDAR 2007. Vol. 2. pp. 694 –698.

[13] Freeman, I. J., Plimmer, B., 2007. Connector semantics for sketched diagram recognition. In: Proceedings of the eight Australasian conference on User interface - Volume 64. AUIC '07. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 71–78.
URL `http://dl.acm.org/citation.cfm?id=1273714.1273726`

[14] Ghorbel, A., Almaksour, A., Lemaitre, A., Anquetil, E., 2011. Incremental learning for interactive for sketch reconition. Ninth IAPR International Workshop on Graphics RECognition - GREC 2011.

[15] Ghorbel, A., Anquetil, E., Lemaitre, A., 2012. Optimization analysis based on a breadth-first exploration for a structural approach of sketches interpretation. In: Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on. pp. 240–244.

[16] Ghorbel, A., Lemaitre, A., Anquetil, É., 2012. Competitive hybrid exploration for off-line sketches structure recognition. In: ICFHR. pp. 571–576.

[17] Ghorbel, A., Macé, S., Lemaitre, A., Anquetil, E., 2011. Interactive competitive breadth-first exploration for sketch interpretation. ICDAR 2011, 1195–1199.

[18] Hammond, T., Davis, R., 2003. Ladder: A language to describe drawing, display, and editing in sketch recognition.

[19] Hammond, T., Davis, R., 2005. Ladder, a sketching language for user interface developers. Computers & Graphics 29 (4), 518–532.

[20] Hammond, T., O'Sullivan, B., 2007. Recognizing free-form hand-sketched constraint network diagrams by combining geometry and context. Proceedings of the Eurographics Ireland 2007.

[21] Hammond, T., Paulson, B., Oct. 2011. Recognizing sketched multistroke primitives. ACM Trans. Interact. Intell. Syst. 1 (1), 4:1–4:34.
URL http://doi.acm.org/10.1145/2030365.2030369

[22] Hammond, T. A., Davis, R., 2009. Recognizing interspersed sketches quickly. In: Proceedings of Graphics Interface 2009. pp. 157–166.

[23] Hilaire, X., Tombre, K., 2006. Robust and accurate vectorization of line drawings. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 890–904.

[24] Lemaitre, A., Camillerapp, J., 2006. Text line extraction in handwritten document with kalman filter applied on low resolution image. Document Image Analysis for Libraries, International Workshop on 0, 38–45.

[25] Lemaitre, M., Grosicki, E., Geoffrois, E., Preteux, F., 2007. Preliminary experiments in layout analysis of handwritten letters based on textural and spatial information and a 2d markovian approach. In: ICDAR 2007. Vol. 2. pp. 1023 –1027.

[26] Lladós, J., López-Krahe, J., Martí, E., 1997. A system to understand hand-drawn floor plans using subgraph isomorphism and hough transform. Machine Vision and Applications 10 (3), 150–158.

[27] luen Do, E. Y., 2001. Vr sketchpad, create instant 3d worlds by sketching on a transparent window. In: Proceedings of CAAD Futures 2001 (Eindhoven. Kluwer Academic Publishers, pp. 161–172.

[28] Macé, S., Anquetil, E., 2009. Eager interpretation of on-line hand-drawn structured documents: The dali methodology. Pattern Recognition 42 (12), 3202–3214.

[29] Mao, S., Rosenfeld, A., Kanungo, T., 2003. Document structure analysis algorithms: a literature survey. In: Proc. SPIE Electronic Imaging. Vol. 5010. pp. 197–207.

[30] Montreuil, F., Grosicki, E., Heutte, L., Nicolas, S., 2009. Unconstrained handwritten document layout extraction using 2d conditional random fields. ICDAR 2009 0, 853–857.

[31] Plimmer, B., Freeman, I., 2007. A toolkit approach to sketched diagram recognition. In: Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1. BCS-HCI '07. British Computer Society, Swinton, UK, UK, pp. 205–213.
URL `http://dl.acm.org/citation.cfm?id=1531294.1531323`

[32] Sezgin, T. M., Davis, R., 2005. Hmm-based efficient sketch recognition. In: Proceedings of the 10th international conference on Intelligent user interfaces. IUI '05. ACM, New York, NY, USA, pp. 281–283.
URL `http://doi.acm.org/10.1145/1040830.1040899`

[33] Sheraz Ahmed, a. M. L., Weber, M., Dengel, A., 2011. Improved automatic analysis of architectural floor plans. ICDAR 2011, 864–868.

[34] Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. Systems, Man and Cybernetics, IEEE Transactions on SMC-15 (1), 116–132.

[35] Yamamoto, R., Sako, S., Nishimoto, T., Sagayama, S., Oct. 2006. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In: Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft, La Baule (France).